



**LET'S!  
ROCK!**

# ZEND FRAMEWORK 2.0

*Juozas "Joe" Kazinkėnas*

*<http://juokaz.com> / [juozas@juokaz.com](mailto:juozas@juokaz.com) / [@juokaz](https://twitter.com/juokaz)*



**LET'S!  
ROCK!**

# WHO IS THIS GUY?

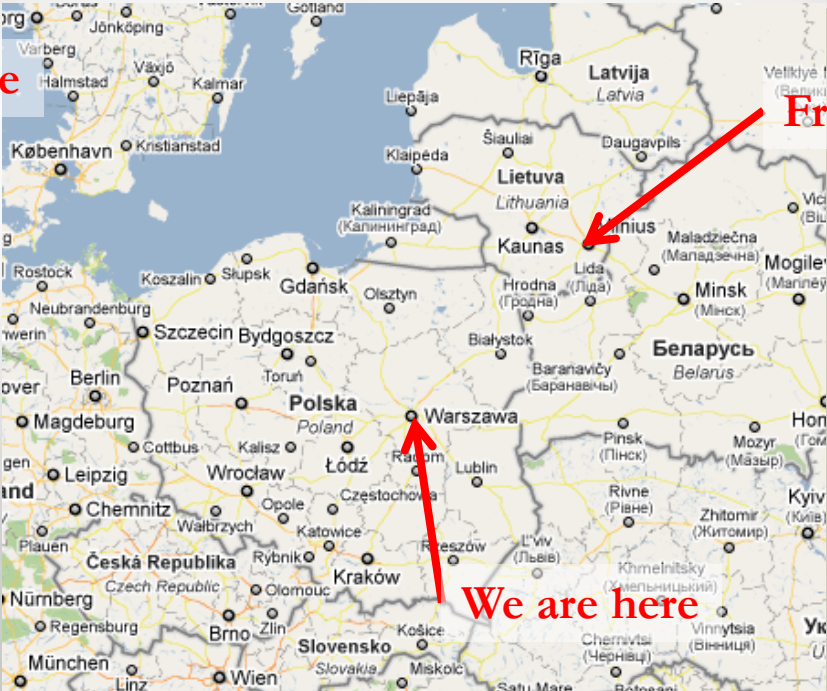
- ❖ Juozas Kaziukėnas
- ❖ People prefer to call me *Joe*
- ❖ Lives in Edinburgh, UK
- ❖ Runs Web Species Ltd. (still can write code)
- ❖ Speaker in PHP conferences all over the world
- ❖ Open source developer (Zend Framework, Doctrine etc.)
- ❖ More info <http://juokaz.com> and [@juokaz](#)

# THE MAP

Now lives here

From Vilnius

We are here





**WS**

**WEB  
SPECIES**

DEVELOPMENT ● CONSULTING ● TRAINING

<http://webspecies.co.uk>

ZEND FRAMEWORK





---

ARE YOU A ZF  
DEVELOPER?



**BE NICE  
OR  
LEAVE  
-THANK YOU**

IF YOU  
ARE  
NOT...





ZEND FRAMEWORK

---

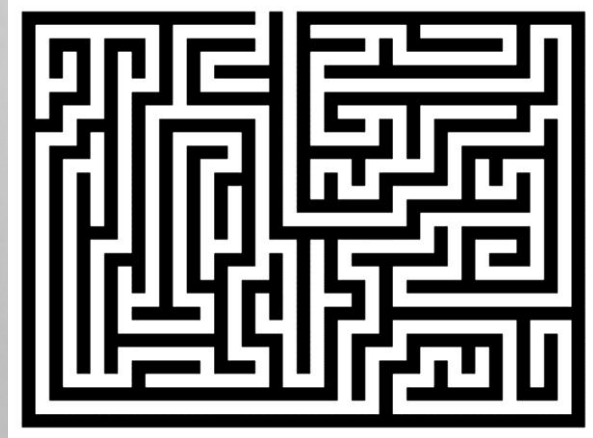




# ZEND FRAMEWORK

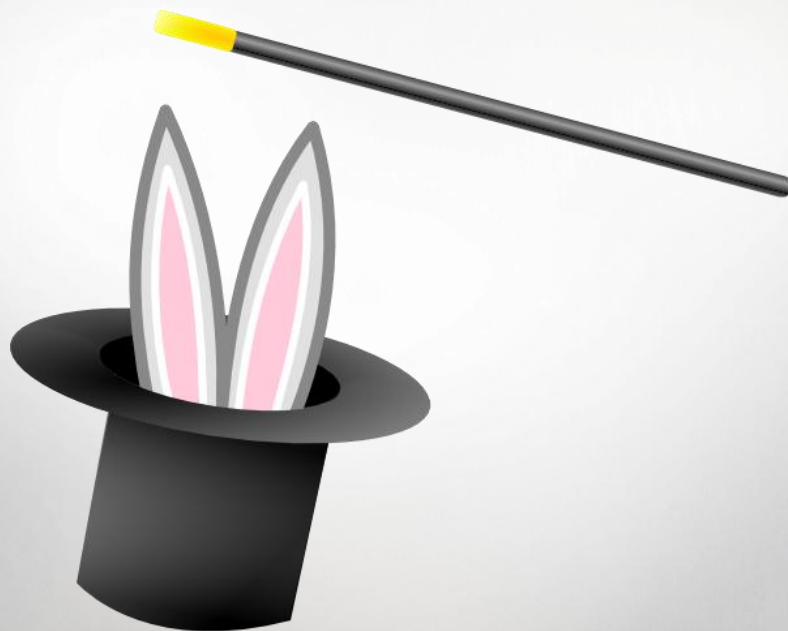
---

- ❖ Journey from October 2005 to today
  - March 2006 – 0.1
  - July 2007 – 1.0
  - Today – 1.11
- ❖ Massive amounts of (high quality) code
- ❖ The most popular PHP framework
  - Developers
  - Business

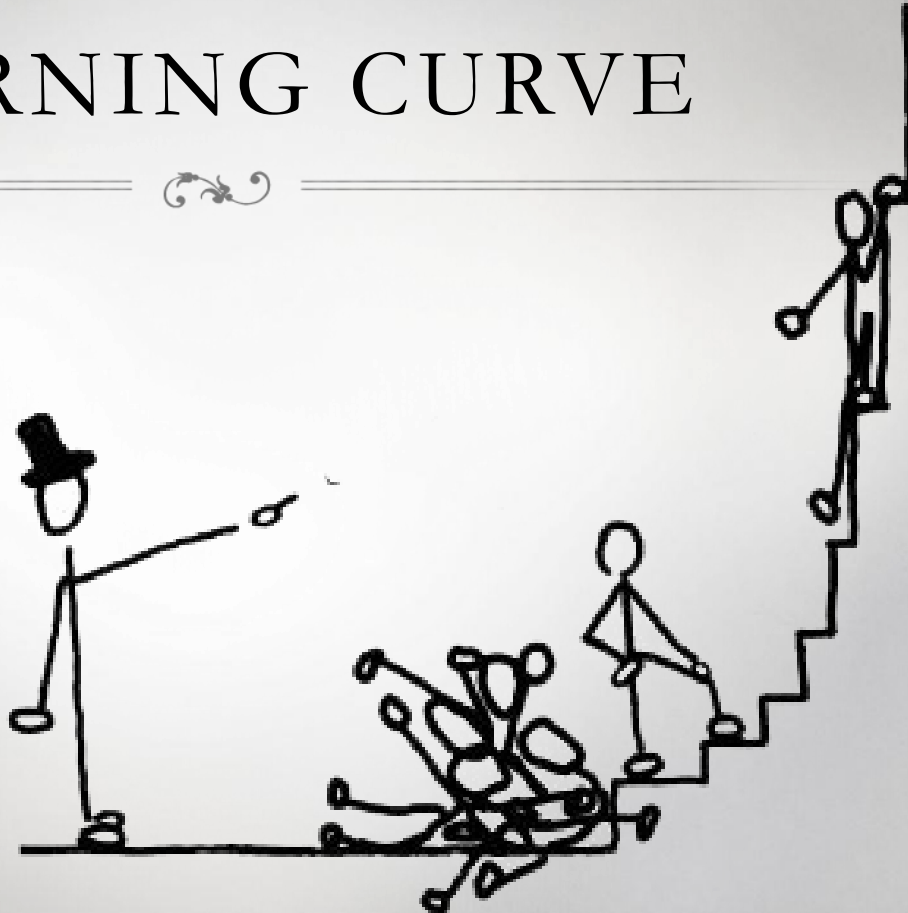


PROBLEMS

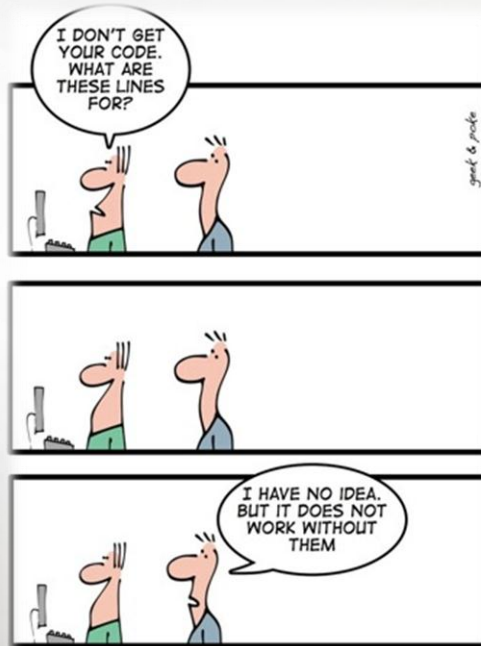
# MAGIC



# LEARNING CURVE



# LEARNING CURVE

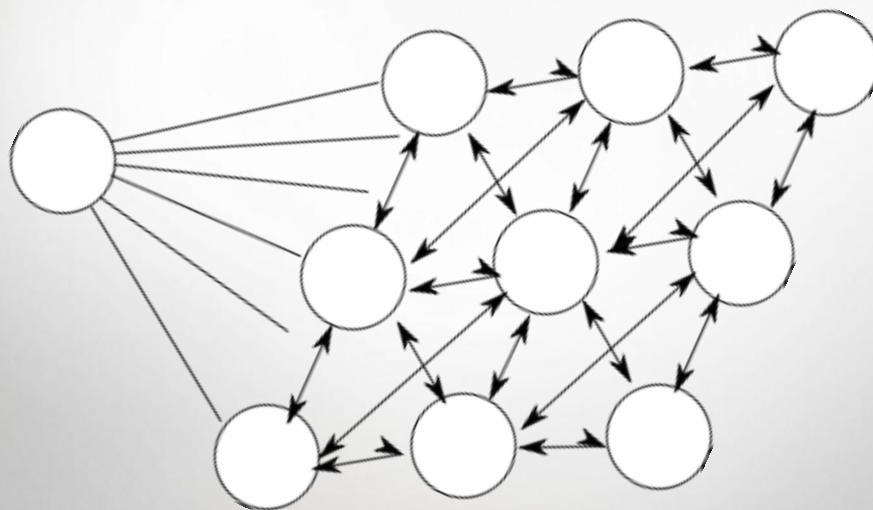


THE ART OF PROGRAMMING - PART 2: KISS

# CONFLICTING



# STILL COUPLED





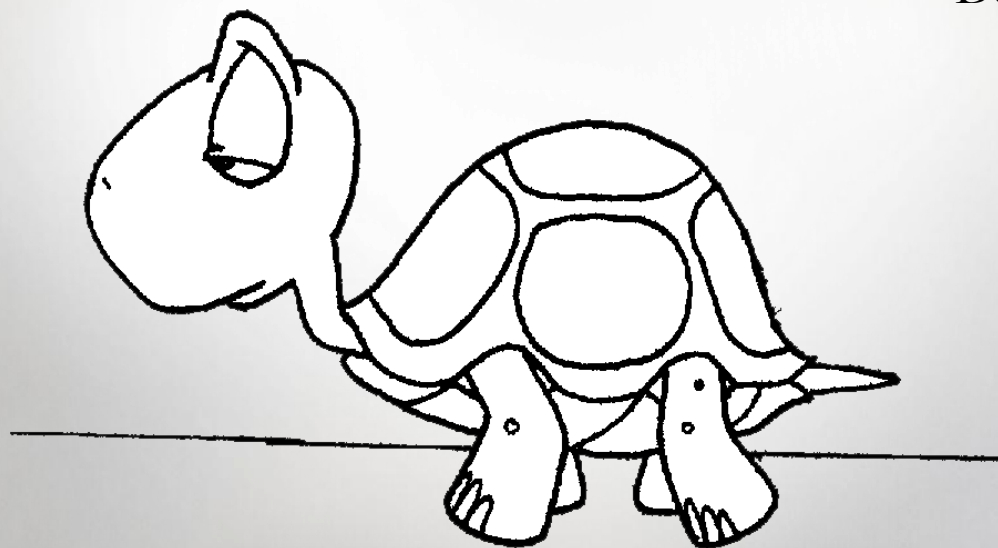


---

MOST IMPORTANTLY...

SLOW\*

Don't forget the \*





2.0

---

ZEND FRAMEWORK 2.0



*Zend Framework 2.0*

---

***THE PRIMARY THRUST OF ZF 2.0 IS TO MAKE A  
MORE CONSISTENT, WELL-DOCUMENTED  
PRODUCT, IMPROVING DEVELOPER  
PRODUCTIVITY AND RUNTIME PERFORMANCE.***



# ZEND FRAMEWORK 2.0

---

- ❖ Development on Git
  - CLA is needed though
- ❖ PHP 5.3 only
  - Old code converted to namespaced
- ❖ Improve, not start from scratch\*



---

TRENDS IN  
*PHP 2011*



# MILESTONES

---

- ❖ Autoloading & Plugin loading
- ❖ Exceptions
- ❖ MVC
- ❖ Testing
- ❖ Documentation
- ❖ Internationalization

# AUTOLOADING

- ❖ `include_path` sucks
- ❖ Use maps
  - Namespace => Path
  - Class name => Path
- ❖ ZF2 ships with class name paths for all components
- ❖ Build your own on deployment





# PLUGINS

- ❖ No more magic
- ❖ Multiple flavors
- ❖ Plugin class => alias
- ❖ Fast

# LOADER EXAMPLES

```
$loader = new \Zend\Loader\StandardAutoloader();

// the path can be absolute or relative below:
$loader->registerNamespace('Zend', '/opt/lib/ZendFramework-2.0.0/library/Zend');

/** TO START AUTOLOADING */
$loader->register();
```

```
$loader->registerMap(__DIR__ . '/../library/Zend/');
```

```
$loader = $broker->getPluginLoader();
$loader->register('foo', 'My\Component\Foo');
$loader->register('url', 'My\Component\Url');

$fooHelper = $broker->load('foo'); // instance of My\Component\Foo
$formHelper = $broker->load('form'); // instance of Zend\View\Helper\Form
$urlHelper = $broker->load('url'); // instance of My\Component\Url
```

# EXCEPTIONS

- ❖ Tree of exceptions in current ZF
  - Exception -> Zend\_Exception -> Zend\_Acl\_Exception -> etc..
- ❖ Using SPL exceptions
- ❖ Component level interfaces
- ❖ Exceptions sub-namespace in component

# EXCEPTIONS EXAMPLE

```
namespace Zend\EventManager\Exception;
use Zend\EventManager\Exception;

class InvalidArgumentException
    extends \InvalidArgumentException
    implements Exception
{}
```

```
namespace Zend\EventManager\Exception;
use Zend\EventManager\Exception;
try {
    $events->trigger('foo.bar', $object);
} catch (InvalidArgumentException $e) {
} catch (Exception $e) {
} catch (\InvalidArgumentException $e) {
} catch (\Exception $e) {
}
```



# MVC

---

- ❖ Many flavors, interface based
- ❖ Modules done right
- ❖ Controllers as services
- ❖ Form & View cleanup

# DEPENDENCY INJECTION

- ❖ Zend\_Registry
- ❖ Service Locator
- ❖ Dependency injection container
  - Java
- ❖ Type hinting in application, plus easier testing

# DIC EXAMPLE

```
$db = new Definition('My\Db\Adapter\Sqlite');
$db->setParam('name', __DIR__ . '/../data/db/users.db');

$mapper = new Definition('My\Mapper\Db');
$mapper->addMethodCall(
    'setAdapter', array(new Reference('db')));

$service = new Definition('My\Resource\Users');
$service->setParam('mapper', new Reference('mapper'));

$di = new DependencyInjector;
$di->setDefinitions(array(
    'db'      => $db,
    'mapper' => $mapper,
    'users'  => $service,
));

$users = $di->get('users'); // My\Resource\Users
```

AT LEAST 200% FASTER







# MORE...

---

- ❖ Documentation
- ❖ Configuration
- ❖ Events
- ❖ Interfaces
- ❖ Community Review team
- ❖ Beer (?)



WHEN?

# WHEN?

- ❖ MVC milestone by the end of May
- ❖ PR release following MVC completion
- ❖ *Beta* in summer
- ❖ Stable by the end of this year

# LET'S ROCK

- ❖ Work in progress
  - MVC milestone in progress
- ❖ Will be faster, easier and overall better
- ❖ Final release this year
- ❖ **Contribute!**



# THANK YOU!

❖ Please leave feedback <http://joind.in/3202>

❖ Contact me on:

- <http://juokaz.com>
- [juozas@juokaz.com](mailto:juozas@juokaz.com)
- Twitter: [@juokaz](https://twitter.com/juokaz)

\* Images from flickr.com and images.google.com